



Experiment 6

Student Name: Lokesh Gupta

Branch: BE-CSE

Semester: 6th

**Subject Name: Advanced Programming
Lab-2**

UID: 22BCS16079

Section/Group: FL_IOT-602/A

Date of Performance: 11/03/25

Subject Code: 22CSP-351

1. Implementation/Code:

[Convert Sorted Array to Binary Search Tree](#)

```
class Solution {
public:
    TreeNode* sortedArrayToBST(vector<int>&
nums) {
        return helper(nums, 0, nums.size() - 1);
    }

    TreeNode* helper(vector<int>& nums, int left,
int right) {
        if (left > right)
            return nullptr;
        int mid = left + (right - left) / 2;    // Middle
element
        TreeNode* root = new
TreeNode(nums[mid]); // Create root node

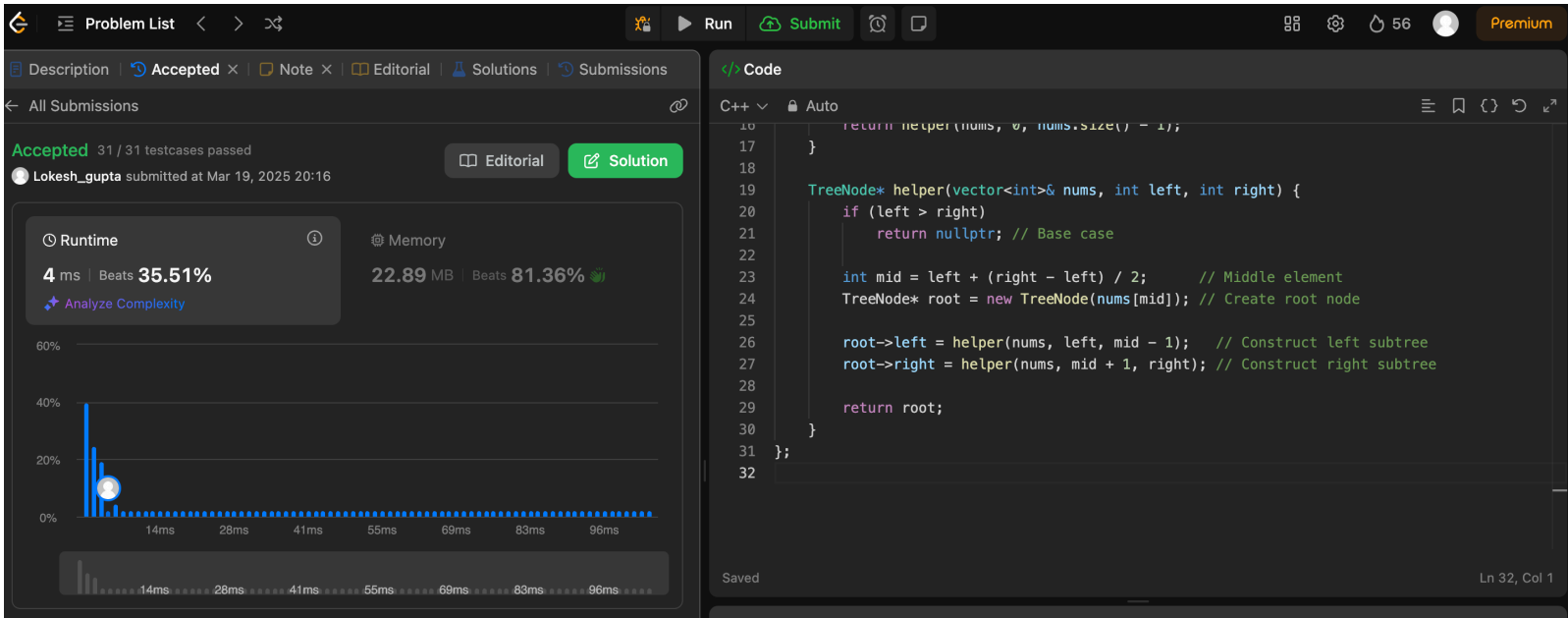
        root->left = helper(nums, left, mid - 1); //
Construct left subtree
        root->right = helper(nums, mid + 1,
right); // Construct right subtree

        return root;
    }
};
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.



Number of 1 Bits

```
class Solution {
public:
    int hammingWeight(int n) {
        int count=0;

        while(n){
            if(n%2 == 1){
                count++;
            }
            n=n/2;
        }
        return count;
    }
};
```



Discover. Learn. Empower.

Problem List < > Run Submit

Description Accepted Note Editorial Solutions Submit

All Submissions

Accepted 598 / 598 testcases passed

Lokesh_gupta submitted at Mar 19, 2025 20:28

Editorial Solution

Runtime 0 ms | Beats 100.00% | Memory 8.25 MB | Beats 47.36%

Analyze Complexity

Code

```
1 class Solution {
2 public:
3     int hammingWeight(int n) {
4         int count=0;
5
6         while(n){
7             if(n%2 == 1){
8                 count++;
9             }
10            n=n/2;
11        }
12        return count;
13    }
14};
```

Sort an Array

```
class Solution {
```

```
public:
```

```
    vector<int> sortArray(vector<int>& nums) {
        mergeSort(nums, 0, nums.size() - 1);
        return nums;
    }
```

```
    void mergeSort(vector<int>& nums, int left, int right) {
```

```
        if (left >= right)
```

```
            return;
```

```
        ... while (j <= right)
```

```
            temp.push_back(nums[j++]);
```

```
        for (int k = 0; k < temp.size(); k++)
```

```
            nums[left + k] = temp[k];
```

```
    }
```

```
};
```

Problem List < > < > < > Run Submit < > 56 Premium

Description Accepted x Note x Editorial Solutions Submiss < >

All Submissions

Accepted 21 / 21 testcases passed

Lokesh_gupta submitted at Mar 19, 2025 20:33

Editorial Solution

Runtime 698 ms | Beats 15.52% Memory 295.67 MB | Beats 11.93%

Analyze Complexity

30% 20% 10% 0%

13ms 290ms 566ms 842ms 1119ms 1395ms 1672ms 1948ms

Code

C++ v Auto

```

1 class Solution {
2 public:
3     vector<int> sortArray(vector<int>& nums) {
4         mergeSort(nums, 0, nums.size() - 1);
5         return nums;
6     }
7
8     void mergeSort(vector<int>& nums, int left, int right) {
9         if (left >= right)
10            return;
11
12        int mid = left + (right - left) / 2;
13        mergeSort(nums, left, mid);
14        mergeSort(nums, mid + 1, right);
15        merge(nums, left, mid, right);
16    }
17
18    void merge(vector<int>& nums, int left, int mid, int right) {
19        vector<int> temp;

```

Maximum Subarray

```

class Solution {
public:
    int maxSubArray(vector<int>& nums) {
        int currentmax=nums[0];
        int maxx=nums[0];

        for(int i=1 ; i < nums.size() ; i++){
            currentmax=max(nums[i],currentmax + nums[i]);
            maxx = max(maxx,currentmax);
        }
        return maxx;
    }
}

```

Problem List < > < > < > Run Submit < > 56 Premium

Description Accepted x Note x Editorial Solutions Submissions

All Submissions

Accepted 210 / 210 testcases passed

Lokesh_gupta submitted at Mar 19, 2025 21:00

Editorial Solution

Runtime 3 ms | Beats 20.61% Memory 71.83 MB | Beats 18.42%

Analyze Complexity

100% 75% 50% 25% 0%

1ms 2ms 3ms 4ms 5ms

Code

C++ v Auto

```

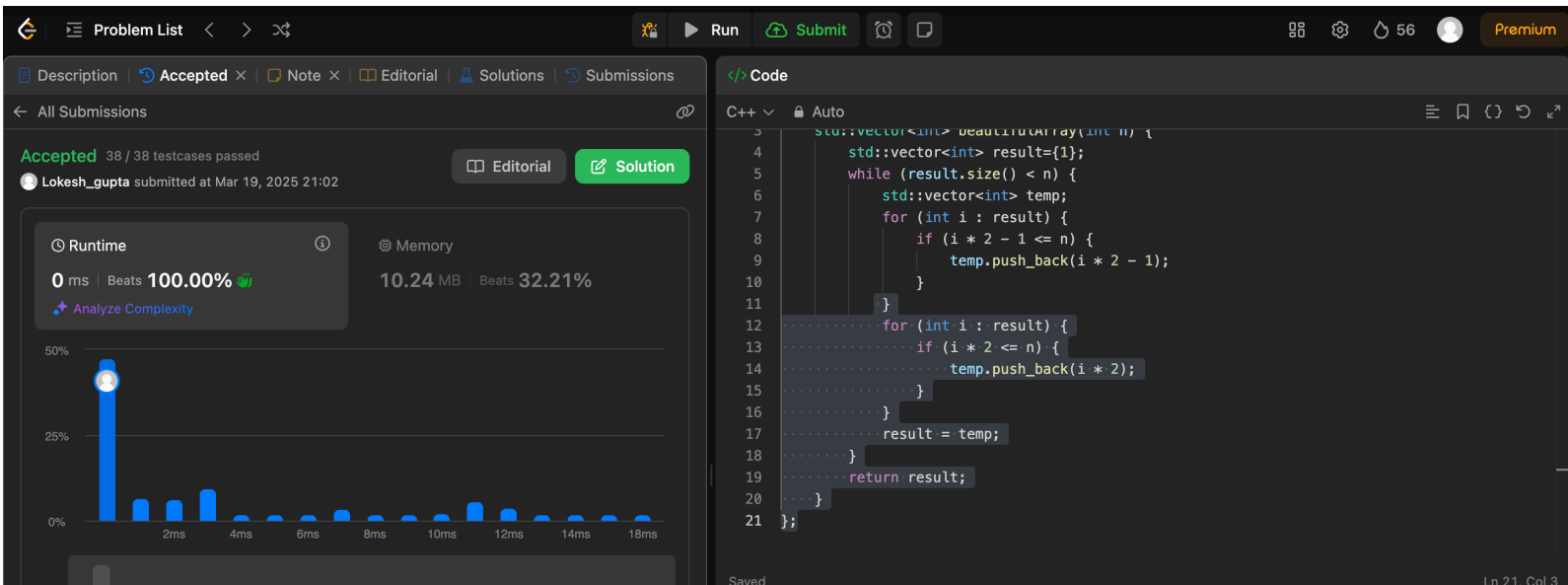
1 class Solution {
2 public:
3     int maxSubArray(vector<int>& nums) {
4         int currentmax=nums[0];
5         int maxx=nums[0];
6
7         for(int i=1 ; i < nums.size() ; i++){
8             currentmax=max(nums[i],currentmax + nums[i]);
9             maxx = max(maxx,currentmax);
10        }
11        return maxx;
12    }
13 }
14

```

Saved Ln 14, Col 3

Beautiful Array

```
class Solution {
public:
    std::vector<int> beautifulArray(int n) {
        std::vector<int> result={1};
        while (result.size() < n) {
            std::vector<int> temp;
            for (int i : result) {
                if (i * 2 - 1 <= n) {
                    temp.push_back(i * 2 - 1);
                }
            }
            for (int i : result) {
                if (i * 2 <= n) {
                    temp.push_back(i * 2);
                }
            }
            result = temp;
        }
        return result;
    }
};
```



Problem List < > ↺


Description | Accepted × | Note × | Editorial | Solutions | Submissions

← All Submissions

Accepted 38 / 38 testcases passed

Lokesh_gupta submitted at Mar 19, 2025 21:02

Editorial Solution

Runtime 0 ms | Beats 100.00%  **Memory** 10.24 MB | Beats 32.21%

Analyze Complexity

50%
25%
0%

2ms 4ms 6ms 8ms 10ms 12ms 14ms 18ms

Code

C++ v Auto

```
1 std::vector<int> beautifulArray(int n) {
2     std::vector<int> result={1};
3     while (result.size() < n) {
4         std::vector<int> temp;
5         for (int i : result) {
6             if (i * 2 - 1 <= n) {
7                 temp.push_back(i * 2 - 1);
8             }
9         }
10        for (int i : result) {
11            if (i * 2 <= n) {
12                temp.push_back(i * 2);
13            }
14        }
15        result = temp;
16    }
17    return result;
18 }
19 }
```

Saved Ln 21, Col 3

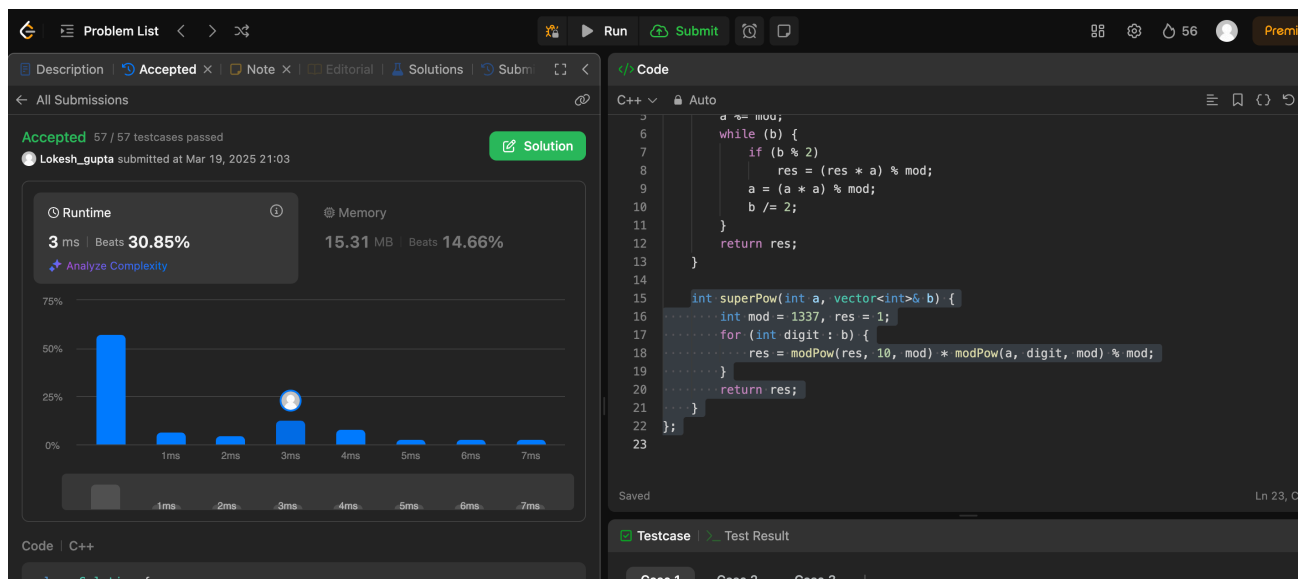


DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Super Pow

```
class Solution {
public:
    int modPow(int a, int b, int mod) {
        int res = 1;
        a %= mod;
        while (b) {
            if (b % 2)
                res = (res * a) % mod;
            a = (a * a) % mod;
            b /= 2;
        }
        return res;
    }
    int superPow(int a, vector<int>& b) {
        int mod = 1337, res = 1;
        for (int digit : b) {
            res = modPow(res, 10, mod) * modPow(a, digit, mod) % mod;
        }
        return res;
    }
};
```



The Skyline Problem

```
class Solution {
```

```
public:
```

```
vector<vector<int>> getSkyline(vector<vector<int>>& buildings) {
```

```
    vector<tuple<int, int, int>> events;
```

```
    for (auto& b : buildings) {
```

```
        events.push_back({b[0], -b[2], b[1]});
```

```
        events.push_back({b[1], b[2], 0});
```

```
    }
```

```
    sort(events.begin(), events.end(), [](auto a, auto b) {
```

```
        if (get<0>(a) != get<0>(b))
```

```
            return get<0>(a) < get<0>(b);
```

```
        return get<1>(a) < get<1>(b);
```

```
    });
```

```
    vector<vector<int>> res;
```

```
    priority_queue<pair<int, int>> pq;
```

```
    pq.push({0, INT_MAX});
```

```
    int prev = 0;
```

```
    for (auto& e : events) {
```

```
        int x, h, r;
```

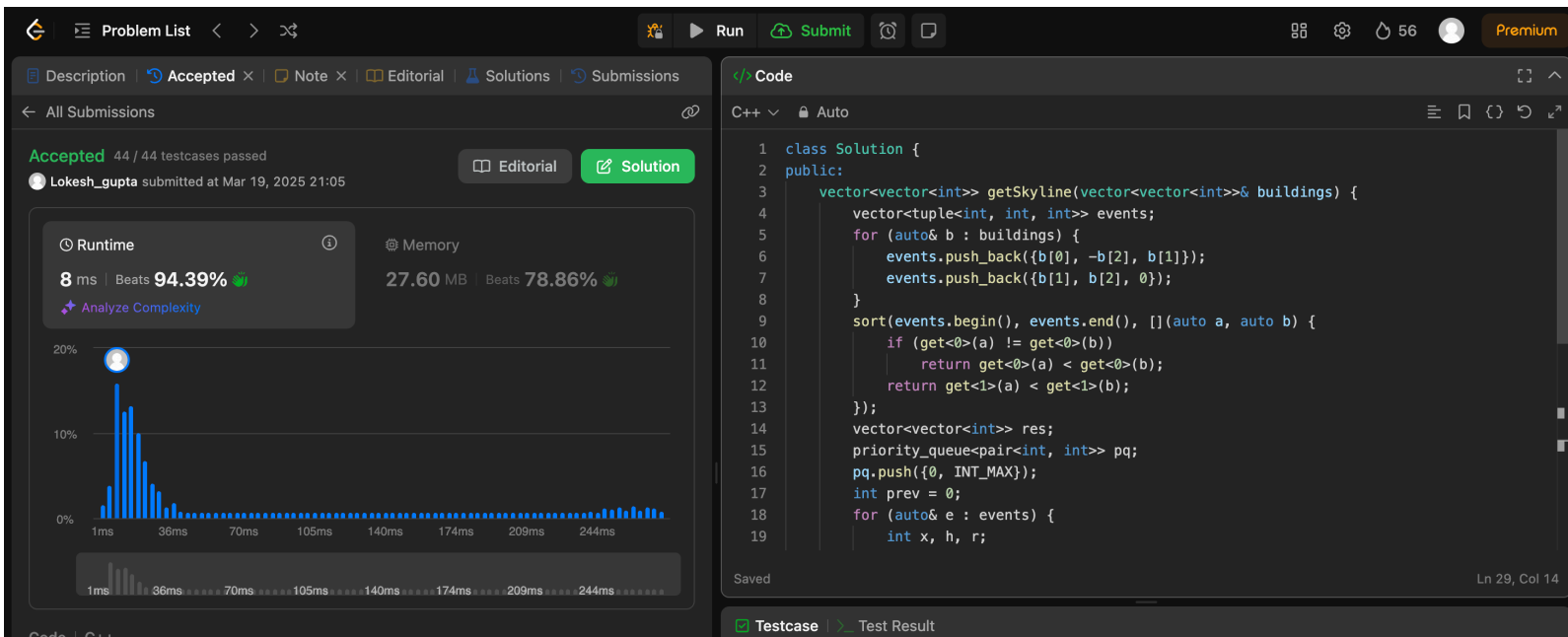
```
        tie(x, h, r) = e;
```

```
        return res;
```

```
    ...
```

```
    }
```

```
};
```



The screenshot shows a coding platform interface with the following components:

- Problem List:** A sidebar on the left showing the problem list with a search bar and filters.
- Code Editor:** A central area with a dark theme, showing the C++ code for the Skyline Problem solution. The code is as follows:


```
1 class Solution {
2 public:
3     vector<vector<int>> getSkyline(vector<vector<int>>& buildings) {
4         vector<tuple<int, int, int>> events;
5         for (auto& b : buildings) {
6             events.push_back({b[0], -b[2], b[1]});
7             events.push_back({b[1], b[2], 0});
8         }
9         sort(events.begin(), events.end(), [](auto a, auto b) {
10             if (get<0>(a) != get<0>(b))
11                 return get<0>(a) < get<0>(b);
12             return get<1>(a) < get<1>(b);
13         });
14         vector<vector<int>> res;
15         priority_queue<pair<int, int>> pq;
16         pq.push({0, INT_MAX});
17         int prev = 0;
18         for (auto& e : events) {
19             int x, h, r;
```
- Runtime and Memory Analysis:** A section on the left showing the execution results:
 - Runtime:** 8 ms, Beats 94.39%.
 - Memory:** 27.60 MB, Beats 78.86%.
 - A bar chart showing the distribution of runtime times across different test cases.
- Testcase Results:** A section at the bottom showing the results of individual test cases.